НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«До захисту допущено»
В.о. завідувача кафедри

_____М.В.Грайворонський
(підпис)
“___”_____2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

на тему: Методи стеганоаналізу у Web-застосунках

Виконав: студент _4_ курсу, групи ФБ-51
(шифр групи)

Філяєв Максим Владиславович _____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник доцент кафедри інформаційної безпеки, к.е.н Ткач В.М. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (посада, вчене звання,науковий ступінь, прізвище, ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)
«__»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту

Філяєву Максиму Владиславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи стеганоаналізу у Web-застосунках
науковий керівник роботи: Ткач Володимир Миколайович, кандидат економічних наук, доцент кафедри ІБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «_____» 2019 р. № _____

2. Термін подання студентом роботи 10 червня 2019 р.

3. Вихідні дані до роботи

Два стеганографічні алгоритми:

1.Перший опрацьовує дані з веб сторінки,

представляючи їх в зручному для класифікатора

вигляді;

2.Другий використовує статистичні дані зі

сторінки для виявлення стеганографії

4. Зміст роботи

1. Дослідити стеганографічні та стеганоаналітичні алгоритми у web-застосунках;

2.Дослідити підхід HPAT для приховання повыдомлень у web-застосунках ;

3.Проаналізувати існуючі рішення у вигляді алгоритмів виявлення HPAT;

4.Розробити нові методи виявлення застосування HPAT у web-застосунках.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): 1)ілюстрації, що містять псевдокод стеганографічних алгоритмів, схем; 2)стеганоаналітичні алгоритми для виявлення HPAT; 3)зображення формул для обрахунку статистичних показників; 4)схема класичної стеганосистеми.

6. Дата видачі завдання 01.02.2019

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Обрання теми	17.09.18-23.09.18	
2	Опрацювання літератури	24.09.18-03.02.19	
3	Написання першого розділу	04.02.19-11.03.19	
4	Написання практичної частини	11.03.19-24.04.19	
5	Написання другої частини	25.04.19-07.05.19	
6	Написання третьої частини	08.05.19-15.05.19	
7	Оформлення дипломної роботи	16.05.19-06.06.19	
8	Захист	19.06.19	

Студент _____ Філяєв М.В.
(підпис) (ініціали, прізвище)

Науковий керівник роботи _____ Ткач В.М.
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Метою дипломної роботи є дослідження існуючих методів стеганографії у web-застосунках та їх стеганоаналізу. Web-застосунки є зручним об’єктом для приховання повідомлення, оскільки задовільняють багатьом необхідним для реалізації стеганографії умовам. Цей факт використовується зловмисниками для реалізації своїх намірів, тому на даний момент існує гостра потреба у ефективних інструментах виявлення стеганографії у web-застосунках. У роботі розглядаються існуючі методи реалізації стеганографії та відповідні методи стеганоаналізу. Особлива увага відведена стеганографічному методу, який працює на основі зміни положення атрибутів у тегах - для його виявлення запропоновано два алгоритми. Загальний обсяг роботи: 63 сторінки, 20 ілюстрацій, 7 таблиць та 24 бібліографічних джерел.

Ключові слова: стеганографія, стеганоаналіз, HTML, цифрова стеганографія.

ABSTRACT

The purpose of the thesis is to study existing methods of steganography and steganoanalysis in web-applications. Web applications are a convenient object for concealing a message, since they satisfy many of the requirements for the implementation of steganography. This fact is used by criminals to reach their goals, so there is an urgent need for effective tools for detecting steganography in web applications at the moment. The paper considers existing methods for the implementation of steganography and the corresponding methods of steganoanalysis. Special attention is paid to the steganographic method, which works by changing the position of the attributes in the tags - two algorithms are proposed for detection. Total volume of work: 63 pages, 20 illustrations, 7 tables and 24 bibliographic sources.

Key words: steganography, steganoanalysis, HTML, digital steganography.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

SV - Support Vector Machine

LSB - Least Significant Bit

NN – Neural Network

HTML - HyperText Markup Language

HCCL - Hiding by Changing the Case of the Letters

ICS - Invisible Characters Steganography

HPAT - Hiding by Permuting Attribute of Tag

AUC - Area Under Curve

SD - Standard Deviation

TP - True Positive

TN - True Negative

FP - False Positive

FN - False Negative

ВСТУП

Людство користувалося стеганографічними методами ще на зорі свого існування, проте класифікація та стрімкий розвиток цієї галузі відбувся у 20 столітті, і не зупиняється у наш час.

Із виникненням Інтернету ми отримали безліч переваг та можливостей, яких не мали раніше. Коммунікація без обмежень, розвиток бізнесу та новий рівень свободи – все це стає доступним разом із інтернет-з'єднанням. Проте незважаючи на величезну кількість плюсів, Інтернет має один дуже суттєвий недолік – він є діючою та дуже зручною платформою для реалізації зловмисних намірів.

Суттєву роль зіграла поява Інтернету і для стеганографії – з'явилися нові напрями як для приховування повідомлень, так і для їх виявлення. Світова мережа стала причиною виникнення ще одного підрозділу стеганографії – цифрової стеганографії.

Аудио і відео файли, зображення та документи є предметом дослідження цифрової стеганографії. Кожен із цих типів представлення інформації може використовуватися для приховування секретного повідомлення, факт наявності якого знає лише отримувач та відправник. Проте всі перелічені типи інформаційних контейнерів мають цінність для стеганографії лише завдяки своїй поширеності - саме через фізичну неспроможність проаналізувати всю інформацію, яка курсує у світовій мережі за добу.

Цифрова стеганографія передбачає використання ще одного контейнеру – Web-сторінки. Особливості мови HTML та розповсюдженість Web-сторінок роблять останні ідеальним об'єктом для стеганографії, яка в свою чергу може використовуватися для злочинних намірів. Саме тому питання виявлення застосування стеганографії до сторінки залишається досі відкритим.

Метою і завданням даної роботи є дослідження методів стеганоаналізу web-сторінок та розробка ефективних методів виявлення HPAT стеганографії.

Об’єктом дослідження є алгоритми цифрової стеганографії у web-сторінках та принципи їх роботи, а також існуючі методи виявлення застосування цих алгоритмів.

Предметом дослідження є методи стеганоаналізу у web-сторінках.

Методом дослідження є ознайомлення та опрацювання електронних джерел на різних мовах про стеганографічні алгоритми та способи їх виявлення.

Наукова новизна дослідження полягає у тому, що були запропоновані нові методи виявлення HPAT стеганографії у web-сторінках.

1 ОСНОВНІ ПОЛОЖЕННЯ СТЕГАНОГРАФІЇ ТА МЕТОДИ СТЕГАНОАНАЛІЗУ

У цьому розділі представлена інформація про стеганографію і стеганоаналіз, описані основні стеганографічні поняття та визначення .

1.1 Стеганографія та її напрями

Стеганографія - це метод передачі або збереження інформації, при якому збеігається секретність факту передачі чи збереження інформації для всіх, крім відправника та отримувача. Використання стеганографії потребує певного носія або контейнера, який виступає у ролі маскуючого покриття для приховування повідомлення. У цьому випадку прихована інформація є стеганоповідомленням, а дані, у яких ця інформація прихована – стеганоконтейнером. Головним критерієм якісного контейнеру є те, що його використання для стеганографічних цілей не повинно привертати уваги третьої сторони – наявність прихованого повідомлення не повинно бути очевидним або викликати підозри у випадкового спостерігача [1].

На відміну від криптографії, яка захищає конфіденційність інформації шляхом її перетворення, стеганографія направлена на приховування факту присутності важливої інформації. Використання криптографічних перетворень вже вказує на цінність інформації, в той час коли стеганографія дозволяє передавати інформацію, не привертаючи уваги. Проте найбільш надійні системи поєднують в собі криптографію та стеганографія – навіть якщо зберігання чи передача повідомлення була розкрита, аналітик буде змушений подолати криптографічний захист повідомлення для отримання необхідної йому інформації.

Стеганографія використовується як для злочинних, так і правомірних цілей. Прикладом правомірного використання є впровадження прихованих

повідомлень для збереження авторського права чи секретна коммунікація між органами правопорядку. Інша сторона стеганографії, кримінальна, відома як зручний метод для контрабанди даних (в тому числі і для цілей промислового шпигунства) та таємного спілкування між зловмисниками.

Більш детальний приклад легального використання стеганографії – водяний знак [3]. Придбані в Інтернеті цифрові дані (книги, зображення, тощо) надаються купуючому без видимого водяного знаку, проте при цьому на них пристуній стеганографічний водяний знак. Різниця між ними полягає у інформації, яку в собі несе стеганографічний знак – такий спосіб маркування є підтвердженням авторства та може використовуватися для протидії цифровому піратству, причому наявність такого знаку непомітна для користувача. Якісний стеганографічний водяний знак є стійким до різних форм спотворень даних - обертання, стиснення і кадрування (якщо мова йде про зображення).

Для зловмисника використання стеганографічних інструментів є привабливим, оскільки:

- зберігається секретність персони відправника та отримувача.
- з’являється можливість не використовувати криптографію, забезпечиючи при цьому секретність повідомлення.

Стеганографію можна класифікувати за різними параметрами та способами представлення. Проте якщо розглядати комп’ютерну стеганографію, виникнення якої можна вважати недавньою подією, можливо виділити такі основні напрями для класифікації:

- стеганографія файлових систем – використовує особливості зберігання файлів різними операційними системами. Вона включає створення спеціальних розділів пам’яті для приховування інформації [4], а також маскування програмного забезпечення.
- цифрова стеганографія – ця категорія базується на застосуванні стеганографії до цифрових медіа, таких як: зображення, відео, аудіо та текстові файли. У цьому напрямі знаходиться і стеганографія HTML сторінок.

- стеганографія мереж – передбачає використання мережевих протоколів для таємного обміну інформацією. Структура та принцип дії мережевих забезпечує дві суттєві характеристики, якими повинен володіти якісний стеганоконтейнер - широке поширення і можливість модифікації без візуально помітної поведінки. Пошкодження деяких пакетів і їх повторна передача передбачена мережевими протоколами, що може бути використано для приховування повідомлення без виникнення підозри [5].

1.2 Модель стеганосистеми

На рисунку 1 зображена класична модель стеганосистеми. Модель передбачає існування двох людей – відправника та отримувача, які відокремлені один від одного та мають можливість спілкуватися тільки через один канал зв’язку. Існуючий канал контролюється третьою особою, а секретність інформації забезпечується спеціальним ключем – стеганоключем. Відправник та отримувач повинні обмінюватися повідомленнями таким чином, щоб не привернути уваги сторонньої особи, яка володіє каналом. При цьому передбачається, що стороння особа є стеганоаналітиком і має можливість модифікувати повідомлення.

Надсилач повинен використати певний метод кодування для приховання повідомлення в об'єкті-носії (стеганоконтейнері). Отримувач використовує стего-ключ для декодування та вилучення повідомлення із стеганоконтейнера. Використання стего-ключа в цій моделі можна вважати аналогом використання ключів шифрування в криптографії.

Секретність прихованого повідомлення залежить від вибору стего-ключа який розподіляється між відправником і одержувачем. Враховуючи цей факт, необхідне існування протоколу стеганографії, з яким попередньо погодились і надсилач, і отримувач.

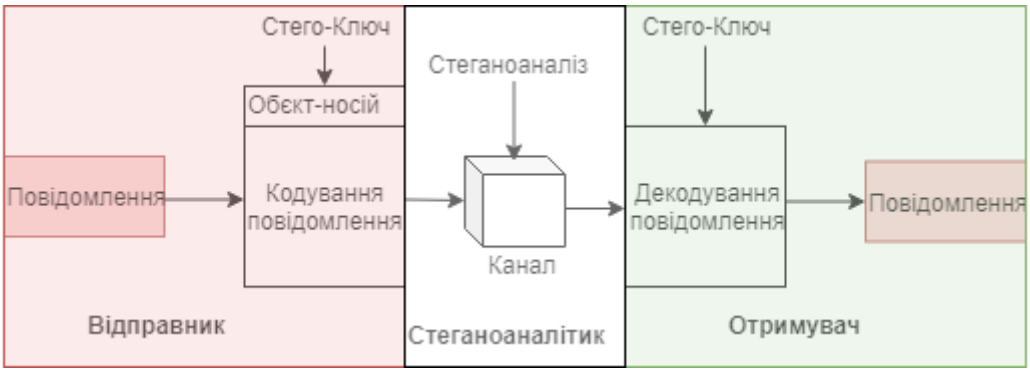

Рисунок 1.1 - Модель стеганосистеми

У класичній моделі стеганосистеми передбачено використання стего-ключа, проте стеганографічні протоколи можуть класифікуватися відповідно до присутності цього ключа у системі.

- Чиста стеганографія: стеганографія, в якій не використовується стего-ключ. Іншими словами, для обміном повідомленнями необхідними є лише алгоритми вбудування та вилучення. У цьому випадку безпека стеганосистеми переважно залежить від її секретності. Такий підхід є основою багатьох сучасних стеганосистем (серед них системи із використанням HTML-стеганографії).
- Стеганографія секретного ключа - для вбудовування і вилучення прихованого повідомлення використовується секретний ключ (відомий тільки відправнику і одержувачу). Принцип дії таких систем нагадує роботу криптографічних систем.
- Стеганографія публічних ключів – у таких стеганосистемах необхідні пара ключів – публічного та приватного. Відкритий ключ використовується для вбудування прихованого повідомлення у контейнер, а закритий ключ застосовується для відновлення прихованого повідомлення.

Класична модель дозволяє визначити деякі важливі поняття, такі як “контейнер” чи “повідомлення”:

- контейнер – інформація, яка використовується як покриття для повідомлення.

- повідомлення – інформація, представлена у вигляді тексту, зображення, аудио чи відео файлу.
- порожній та заповнений стеганоконтейнер – контейнер без повідомлення та із прихованою інформацією відповідно [6].

1.3 Цифрова стеганографія

Цей напрям є підрозділом комп'ютерної стеганографії і базується на приховуванні чи додаванні інформації до цифрових об'єктів. Застосування цифрової стеганографії передбачає використання надлишковостей, характерних для медіа даних різного типу (фото, текст, тощо). Присутність надлишковостей дозволяє приховати у них інформацію, заміняючи останні на повідомлення. Так, маніпулюючи кольоровою палітрою зображення, можливо закодувати у ньому текстове повідомлення [7]. Надлишковість присутня і у інших форматах представлення інформації, завдяки чому їх використовують у якості стеганоконтейнерів.

Модифікація файлів-контейнерів призводить до змін, проте використання ефективних стеганографічних алгоритмів робить ці зміни непомітними не тільки для людини, а й для стеганоаналітичних методів.

Цифрова стеганографія існує через поширеність медіа файлів – вся світова мережа складається із web-сторінок, зображень, відеофайлів та інших потенційних контейнерів. Фізично неможливо перевірити таку кількість інформації, тому перелічені формати використовуються зловмисниками для досягнення своїх цілей.

1.3.1 Цифрова стеганографія зображень

Зображення є одним із найпоширеніших типів стеганоконтейнерів. Використання цього формату для вбудування повідомлень має багато переваг:

- інтернет містить незліченну кількість зображень, які щоденно додаються та оновлюються – завантаження зображення-стеганоконтейнера не викличе підозри при його звичайному перегляді.
- ємність зображень дозволяє передавати значну кількість інформації. При цому існує гнучкусть, у вигляді вибору між ємністю контейнера та непомітністю модифікацій стеганоконтейнеру.
- врахування особливостей людської біології(зору) при розробці стеганоалгоритмів може позитивно вплинути на приховання модифікацій у зображенні [8].

Цифрова стеганографія у зображеннях використовується для підтвердження авторського права, зокрема для додавання підписів і цифрових відбитків, та забезпечення цілісності зображення. Методи стеганографії зображень можна класифікувати за принципом роботи алгоритмів.

- Просторові методи приховування повідомлення. Цей тип передбачає приховання повідомлення у просторовій області стеганоконтейнеру. Для кодування повідомлення використовуються найменш значущі біти - молодші біти пікселів зображення-контейнеру. Відомий алгоритм LSB (Least Significant Bit) містить у назві принцип своєї дії [9].
- Частотні методи. Цей тип методів реаліюазується використанням ДКП (дискретного косинусного перетворення) та модифікації LSB. Методи цієї групи є більш стійкими до стеганографічних атак, таких як спотворення контейнеру чи аналіз частотних показників зображення. Деякі із модифікацій алгоритмів здатні витримувати комплексні стеганоатаки і важко піддаються розкриттю.

Стеганографія зображень уже достатньо довгий час успішно використовується як зловмисниками, так і силами правопорядку. Великий потенціал та гнучкусть методів стеганографії зображень сприяє створенню нових стеганографічних та стеганоаналітичних алгоритмів.

1.3.2 Стеганографія текстових файлів

Розповсюдження текстової інформації є одним з найдавніших засобів масової інформації, який може бути використаний як стеганографічний контейнер. Велика кількість текстової інформації в Інтернеті дозволяє стеганографам використовувати цей формат представлення даних у якості носія прихованих повідомлень.

Існує багато прикладів реалізації алгоритмів застосування тексту у якості сховища даних, проте у області текстової стеганографії існує суттєва проблема. Текстові файли мають дуже незначну надмірність у порівнянні із зображеннями, відео та аудио файлами. Наразі існує багато модифікацій, які дозволяють збільшити надлишковість тексту не змінюючи вигляд документу [9]. Опираючись на роботу Gary C. Kessler [11], методи текстової стеганографії можна класифікувати на:

- методи, які базуються на форматі та характеристиках тексту для приховування даних – форматування, знаки пунктуації, алфавіт.
- лінгвістичні методи, які використовують синтаксичні і семантичні особливості мов.
- методи, які опираються на приховування даних з урахуванням статистично «природнього» вигляду тексту. Реалізація таких алгоритмів орієнтована на збереження частот букв та слів тексту.

Разом з цим можливо розділити текстову стеганографію на більш загальні течії – лінгвістичну та технічну. Остання передбачає використання технічних аспектів текстового документу, таких як приховування даних за допомогою

невидимих написів, зміни форми та кольору літер, використання технічних полів та заміна коду символів.

Ціллю лінгвістичної стеганографії є використання складніших методів, орієнтованих на специфічність певної мові та правила правопису - перенесення слів, перенесення рядків, вибіркових помилок. Методи лінгвістичної стеганографії можуть базуватися на шифрах чи жаргоні – інформації, зрозумілої лише для обмеженої групи людей. Найвідомішим є метод нульового шифру, суть якого полягає у дотриманні попередньо встановлених правил для вбудування та вилучення прихованих повідомлень. Приклад: якщо було встановлене правило читання кожної першої букви нового абзацу, то отриманий набір символів буде представлений у вигляді осмисленого повідомлення.

Один із відомих напрямків у лінгвістичній стеганографії - обробка природніх мов (Natural Language Processing). Він базується на виявленні синонімів у мовах та їх використанні для кодування повідомлення.

1.4 Стеганоаналіз цифрової стеганографії

Використання стеганоаналізу визначається як виявлення присутності секретної інформації у зберігаємих данніх або передаваємих повідомлень. У багатьох принципах стеганаліз дуже схожий на криптоаналіз, який застосовується в криптографії. Основна відмінність між стеганоаналізом та криптоаналізом полягає у інформації про секретність повідомлення – криптоаналіз застосовується до повідомлень з цінною інформацією, в той час як стеганоаналіз починається із підозри про цінність повідомлення. При криптоаналізі вже наявні докази цінності повідомлення – воно захищенне шифруючим алгоритмом. Це одразу виказує цінність даних, проте для отримання секрету необхідне їх розшифрування. Іншими словами, головною метою стеганоаналізу є відповідь на питання – містить файл-контейнер приховане повідомлення чи ні. Дуже часто стеганоаналіз передбачає не тільки

підтвердження факту наявності прихованої інформації у стеганоконтейнері, а і її вилучення – все залежить від цілей аналітика.

Стеганоаналітик змушений враховувати багато аспектів та проблем при аналізі стеганоконтейнеру.

- Приховані дані можуть розташовуватися в послідовних (специфічних) місцях чи випадково розподілені по стеганоконтейнеру.
- Секретні повідомлення можуть вбудовуватися в ланцюжок з використанням декількох стеганоконтейнерів.
- Іноді неможливо обробити множину контейнерів. Прикладом для HTML стеганоаналізу є спроба виявити наявність стеганографії у кожній web-сторінці – вирішення цієї задачі наразі є неможливим.
- Стеганоаналіз є ресурсомістким процесом - він потребує попередньої підготовки та часу.

Найбільш відомі методи стеганоаналізу – візуальний та статистичний. Якщо надійність першого залишається під питанням, то другий є значно ефективнішим і передбачає спостереження за процесом заміни надмірностей в контейнері при додаванні повідомлення. Будь-яка модифікація призводить до змін статистичних властивостей оригінального контейнеру, що може вказувати на факт використання стеганографії, і задачою стеганоаналітика є виявлення цих змін [11].

Існують різні способи класифікації стеганоаналізу [12]. Один із багатьох варіантів – класифікація стеганоаналізу, в основі якого знаходяться певні знання аналітика. Це можуть бути знання про природу стеганоконтейнеру, прихованого повідомлення чи оригінального контейнеру або ж використаного алгоритму стеганографії.

Відповідно до цього можна визначити види атак при стеганоаналізі, які наведені у таблиці 1. Очікується, що аналітик володіє інформацією про відмічений у таблиці елемент і будує свою атаку на основі своїх знань. Ефективність атаки збільшується, якщо стеганоаналітик отримує більше інформації про об ’єкт стеганоаналізу.

Якщо стеганоаналітик володіє знаннями про суть прихованого повідомлення – одразу відпадає потреба стеганоаналізу та побудови атаки на стеганоконтейнер.

Таблиця 1.1 - Типи атаки в залежності від знань стеганоаналітика

	Стегано об'єкт	Оригінальний контейнер	Приховане повідомлення	Алгоритм стеганографії
Стегано об'єкт	З			
Контейнер	З	З		
Повідомлення відоме	З		З	
Обраний метод	З			З
Повідомлення	З	З		
Все окрім к.	З	З		З

З таблиці можна побачити, що найскладнішою для стеганоаналітика є реалізація атаки, яка базується лише на знанні наявності використання стеганографії на контейнер. Протилежний випадок – стеганоаналітик володіє всією інформацією, крім самого повідомлення – побудова такої атаки є найлегшим варіантом.

1.4.1 Класифікація методів стеганоаналізу

Більш глобально стеганоаналіз можна поділити на загальний та спрямований [13]:

- спрямований стеганоаналіз (Targeted Steganalysis). Його ціль – виявлення повідомлень у стеганоконтейнерах, причому стеганоаналітик володіє деякою інформацією про застосований алгоритм стеганографії. Знання природи

стеганографічного алгоритму дозволяє ефективно виявляти та вилучати приховані повідомлення..

- загальний (Universal). Методи загального стеганоаналізу використовують у випадку, коли є підозра наявності прихованого повідомлення, проте суть стеганографічного алгоритму залишається невідомою. Загальний стеганоаналіз має меншу точність виявлення, ніж спрямований, проте він дозволяє обробляти більшу кількість підозрілих контейнерів.

Серед двох методів, спрямований стеганоаналіз має вищу ефективність через знання стеганографічного алгоритму. В свою чергу, загальний стеганоаналіз володіє більшою швидкодією і може обробляти значні масиви даних.

Слід зазначити, що якісний стеганоаналіз потребує попереднього отримання та обробки інформації про об’єкт дослідження(стеганоконтейнер). Якщо розглядати цей підхід на прикладі текстового файлу, то стеганоаналітику необхідно провести видалення всіх знаків пунктуації та задати певне маркування. Результатом цих дій буде отримання набору слів, які надалі будуть аналізуватися. До попередньої підготовки може відноситися формування словника найпоширініших символів та слів, виділення слів-ключів чи дослідження лінгвістичних особливостей мови. Такий підхід значно полегшить подальший стеганоаналіз.

Висновки до розділу 1

У цьому розділі було розглянуто основні задачі та напрями стеганографії і стеганоаналізу. Важливу роль для стеганографії грає тип контейнеру, куди приховується повідомлення. Особливості контейнеру використовуються для розробки ефективних алгоритмів, виявлення присутності котрих потребує значних ресурсів.

Попередня обробка та отримання даних при стеганоаналізі є важливим кроком для виявлення застосування стеганографії. Знання природи стеганографічного алгоритму та попередня обробка інформації дозволяють виявляти стеганографію.

2 СТЕГАНОГРАФІЯ ТА СТЕГАНОАНАЛІЗ У WEB-ЗАСТОСУНКАХ

В цьому розділі розглядаються основні методи стеганографії у web-сторінках та способи виявлення застосування цих методів. Також підіймається питання про використання класифікатора як засобу стеганоаналізу.

2.1 Чому можлива стеганографія в HTML файлах

Мова розмітки документів, або HTML (HyperText Markup Language), використовується для створення майже всіх web-сторінок у сучасному Інтернеті. Ідея використання стеганографії у web-сторінках з’явилася практично разом із виникненням самого Інтернету. Існує досить багато стеганографічних методів у web-сторінках, проте серед них можна виділити:

- додавання інформації у мета-теги, які не відображаються для звичайного користувача.
- запис повідомлення у кінець HTML файлу.
- кодування повідомлення у коментарі, які невидимі для користувачів.
- додавання символів табуляції.
- зміна порядку атрибуту у тегах.

Кожен із цих методів використовує певні особливості мови HTML, такі як нечутливість до порядку розташування тегів чи порядок зчитування браузером сторінки. Кожен з перелічених методів не змінює вигляд сторінки для користувача, проте на практиці деякі із них неефективні для стеганографії, оскільки можуть легко розкриватися навіть без використання спеціальних засобів.

Web-сторінки є ідеальними стеганоконтейнерами, оскільки їх використання не викликає підозри через свою поширеність, з їх допомогою

можна передати секретну інформацію не розкривши особу отримувача і, головне – структура HTML документу дозволяє реалізовувати стеганографічні алгоритми.

2.2 Стеганографія невидимих символів

Під час обробки тегів сторінки, браузер не звертає уваги на невидимі символи, такі як символи табуляції, пробілу, та невидимого пробілу (U+0020, U+0009, U+200B відповідно) [14]. Іншими словами, для браузера немає різниці між тегом
 та <br />. Саме це і використовує стеганографія невидимих символів, або Invisible Characters Steganography (ICS). За допомогою цього підходу можливо приховати повідомлення у тегах чи просто додавши зайві символи у кінець файлу – це не призведе до видимих змін у web-сторінці.

Цей вид стеганографії відомий також як стеганографія відкритого простору - Open Space Steganography. Цей метод має як переваги, так і недоліки. Серед переваг можна виділити наявність символів пробілу чи табуляції в будь-якому документі, та їх значну кількість – у великому тексті символ пробілу з’являється статистично частіше, ніж будь-який інший. Недоліком використання ICS є збільшення розміру текстового файлу – звичайного чи webb-сторінки. Якщо говорити про візуальну різницю заповненого стеганоконтейнеру від порожнього (web-сторінка із повідомленням та без), то для звичайний користувач не здатен виявити змін. Техніка ICS може використовувати один або комбінувати такі способи приховування повідомленя:

- зміна відстані між словами або використання символу пробілу разом із невидимими символами пробілу, множення, тощо.
- зміна відстані між абзацами та реченнями.
- додавання пробілів і табуляції у кінець сторінки після тегу </html> - браузер ігнорує інформацію за межами цього тегу.
- зміна інтервалу між параграфами.

Прикладом використання символів табуляції для приховання повідомлення можна обрати її додавання у тег:

Якщо при зчитуванні тегів пара <article> …. </article> умовно відповідає 0, тоді <article> …. </article > умовно відповідає 1. Таким чином виникає можливість змінювати порядок використання тегів. Використовуючи такі маніпуляції, будь-хто може закодувати у web-сторінці посимвольне повідомлення (приховані повідомлення представлені у двійковій формі). Разом із розміром сторінки збільшується і її ємність, а долучення додаткових сторінок дозволяє передати текст достатньо великого розміру.

Існуючі алгоритми передбачають і ситуацію, коли розмір повідомлення перевищує ємність сторінки. У такому використовується поєднання алгоритмів – частина інформації кодується за допомогою тегів, а залишок записується у кінець документу. Сучасні алгоритми ICS спочатку оцінюють розмір контейнеру, а потім обирають спосіб приховання повідомлення [15].

2.3 Стеганографія зміни регістру символів

Стеганографія зміни регістру або принцип Hiding by Changing the Case of the Letters (HCCL) користується невибагливістю мови HTML до зміни регістру. Приклад: при зчитуванні тегу <head> його зміна на </HeAd> може бути представлена як 1010 – зміна стандартного написання не призведе до зміни web-сторінки чи труднощів при ідентифікації тегу.

У якості прикладу використаємо псевдокод алгоритму Суі-Луо, описаного у [18]. Перед визначенням псевдокоду для вищезазначеного алгоритму стеганографії слід ввести такі визначення:

• H - web-сторінка, створена мовою HTML.

• M- бінарні кодовані текстові повідомлення, приховані в H. M = $\{m_1, m_2, m_3, \ldots m_n\}$.

• Q: набір імен тегів у H.

Псевдокод описаний на рисунку 2.1. На вході алгоритм обробляє web-сторінку H та повідомлення M. Кінцевим результатом роботи алгоритму є стегано web-сторінка H^j з прихованим повідомленням M. Алгоритм працює таким чином: якщо біт повідомлення $m_i = 1$, то літера тегу змінює регістр на верхній. В іншому разі нічого не відбувається, і алгоритм переходить до наступної літери тегу. У даній реалізації використовуються усі літери тегів, проте існують алгоритми, які використовують лише перші літери певних тегів.

```
1: Input a webpage H
2: Input a message M
3: Output= a stego webpage H^j
4: Find Q set
5: while M f = 0 do
6:     for each tag T ∈ Q do
7:         if m_i = 1 then
8:             Switch a tag letter to uppercase
9:         else
10:            No switching
11:        end if
12:        M = M − m_i
13:    end for
14: end while
```

Рисунок 2.1 - Псевдокод алгоритму HCCL

Як і у випадку алгоритмів ICS, сучасні алгоритми зміни регістру зазвичай порівнюють довжину повідомлення та ємність контейнеру.

На відміну від ICS, використання HCCL не призводить до збільшення розміру HTML файлу. Проте цей метод стеганографії є дуже очевидним для аналізу. Часте використання зміни верхнього та нижнього регістру в тегах може викликати підозри про наявність прихованого повідомлення у сторінці. Практичне використання HCCL можливе лише для коротких повідомлень у багатосторінкових ресурсах.

2.4 Стеганографія перестановки атрибутів

Стеганографія атрибутів, або метод Hiding by Permuting Attribute of Tag (HPAT) також використовує властивість мови HTML. Як відомо, HTML не чутлива до порядку розміщення атрибутів всередині тегу. Наприклад, будь-які зміни порядку атрибутів тегу не призведуть до зміни змісту чи візуальної модифікації сторінки.

Проте за такої комбінації з’являється 6 способів упорядкування атрибутів для тегу <img/>. Змінюючи порядок їх розтановки у кожному наступному тезі, стає можливим закодувати повідомленя. Чим більше атрибутів має тег, тим більша довжина приховуємого повідомлення.

При використанні HPAT не змінюється розмір HTML файлу та вигляд сторінки. Виявити застосування цього стеганографічного методу зазвичай можна лише за допомогою стеганоаналізу. Серед прикладів реалізації даного методу відомі алгоритми Хуанг-Зонг та Шен [19]. Вони схожі по принципу дії, проте мають певні відмінності. Для опису алгоритмів використовується:

- *H* - HTML web-сторінка.
- *M* - повідомлення, яке буде приховано алгоритмом в контейнері H.
- *N* – число, представлене у вигляді ASCII кодів. Ним представляється M.
- *T* - тег з набором атрибутів в H. T = { $a_1, a_2, a_3, \ldots a_m$ }, де m - кількість атрибутів у тезі (| T | = т).
- *Q* - Набір тегів в H, у яких | T | ≥ 2.

Псевдокод алгоритму Хуанг-Зонг представлений на рисунку 2.2.

```
1: Input a webpage H
2: Input a message M
3: Output= a stego webpage H^j
4: N=large number made of the ASCII codes representing M.
5: M=N
6: Find Q set
7: while M f= 0 do
8:    for each tag T ∈ Q do
9:       M^j=M div m!        (m is the number of T attributes)
10:      p=M mod m!          (p is a number between 0 and m!-1)
11:      Transform p to a permutation
12:      Replace T with the new permutation
13:      M=M^j
14:   end for
15: end while
```

Рисунок 2.2 – Псевдокод алгоритму Хуанг-Зонг

На вхід алгоритму приймається web-сторінка H та повідомлення M. Результатом дії є стеганоконтейнер H^j з прихованим повідомленням, яке записано у зміну позицій тегів.

При реалізації алгоритму Шен двійкове відношення між атрибутами тегу трансформується у двійковий рядок. Шен є більш складним алгоритмом, і для його роботи використовуються функції:

- функція SubMessage - використовується для пошуку під-повідомлення sm у M, при цьому довжина | sm | = | T | - 1.
- функція BinaryString - перетворює | T | - 1 атрибути у двійковий рядок BS.
- Tt - набір перестановок BS.
- функція Trunc – відділяє sm від M.

Псевдокод для алгоритму Шен зображений на рисунку 2.3:

```
1: Input a webpage H
2: Input a message M
3: Output= a stego webpage H'
4: Find Q set
5: while M f= 0 do
6:     for each tag T ∈ Q do
7:         sm=SubMessage(m, |T | −1)
8:         BinaryString=transform |T | − 1 attributes to a binary string BS
9:         Find G a set of BS permutations
10:        p=permutation in G that equals to sm
11:        Replace T with p.
12:        M=Trunc(M,sm)
13:    end for
14: end while
```

Рисунок 2.3 – псевдокод алгоритму Шен.

Як і у Хуанг-Зонг, на вхід алгоритму приймається web-сторінка H та повідомлення M. На перших кроках алгоритм обирає sm і генеруює BS для представлення взаємозв'язку між атрибутами тегів. BS генерується з використанням словникового порядку перших букв атрибутів. Якщо перша літера атрибута a_1 знаходиться в алфавітному порядку раніше першої літери атрибута a_2, то біт у BS дорівнює 0. У іншому випадку біт дорівнює 1. Далі генерується множина tt, що включає всі перестановки BS. Перестановки виду tt = sm використовуються для заміни T.

Представлені алгоритми можуть приховувати достатньо великі повідомлення, зберігаючи при цьому секретність. Їх більш складні реалізації передбачають оцінку ємності сторінки.

2.5 Стеганоаналіз у HTML файлах

Кожному із описаних механізмів стеганографії протиставляється окремий стеганоаналітичний підхід, який базується на знанні принципу роботи алгоритму і виявленні змін у web-сторінці.

2.6 Стеганоаналіз HPAT

Серед відомих запропоновано два загальні метода виявлення HPAT: Полака-Котульського [21] та Луі-Шенг[20].

Робота Полака-Котульського базується на ідеї ідентифікації домінуючої впорядкованої пари атрибутів.

Запропонована методика має статистичний характер - для кожної пари атрибутів підраховується кількість випадків, коли перший атрибут з'являється перед другим і навпаки. Припускається, що web-сторінка контейнер має більш хаотичне упорядкування атрибутів ніж звичайна. Алгоритм враховує набір m тегів $T = \{t_1, t_2, t_3, \ldots t_m\}$ і передбачає, що кожен тег t_i має k_i входжень. Кожен тег $t_{i,j}$ має набір атрибутів, які з ним пов'язані:$A_{i,j} = \{a_1, a_2, \ldots\}$ де і - ідентифікатор тегу, а j - номер входження тегу (j ≤ k). Якщо тег t_i має пару атрибутів a_x та a_y впорядковану так, що a_x переходить a_y в більшості випадків зчитування t_i, то значення $R_{ij}(a_x, a_y) = 1$. В іншому разі значення R дорівнює 0.

Якщо розподіл R \ t кожної пари атрибутів a_x, a_y та a_y, a_x, не відрязніється, то $R_{ij}(a_x, a_y) = 1$. Вводиться спеціальне значення W, що обчислюється за допомогою рівняння, зображеного на рисунку 2.4. В рівнянні обчислюється частку пар атрибутів a_x і a_y(зустрічається з різним порядком розташування атрибутів) від загальної кількості сполучень C.

$$\forall x \forall y \; x < y : W = 1 - \frac{\sum_{n=1} m \sum_{i=1} k \; Rn,i(ax,ay)}{C}$$

Рисунок 2.4 – Формула обрахунку W

Коли величина W приймає значення близьке до 1, свідчить про те, що пара атрибутів a_y та a_x зустрічалася частіше, ніж атрибути пар a_x та a_y. У такому разі значення W відрізняється від передбачуваного, що може свідчити про зміну структури web-сторінки. При такому підході W виступає як деякий ідентифікатор використання стеганографії, який має порогове значення.

Значення W повинне бути константою при розгляді будь-якої web-сторінки, де приховане повідомлення передається безперервно, та змінною, якщо приховане повідомлення передається тільки протягом визначеного періодом часу.

Алгоритм Луі-Шенг ґрунтується на ідеї навчання моделі класифікації. Це може використовуватися для виявлення присутності застосування HPAT стеганографії. Для роботи з алгоритмом використовувалася класифікація на основі SVM. Вектори ознак сформовані із використанням таких статистичних даних: відстанню між еталонною середньою позицією атрибутів і вибіркою середнього значення позиції атрибутів стеганосторінки та дисперсії позицій атрибутів.

2.6.1 Інтелектуальний аналіз даних

У підході для стеганоаналізу HPAT доцільно використовувати інтелектуальний аналіз даних, який є підрозділом машинного навчання. Цей метод відноситься до попередньої обробки інформації і полягає у виділенні та виборі корисної для стеганоаналітика інформації, шляхом аналізу значної кількості даних.

Інтелектуальний аналіз даних дозволяє отримати якісний набір даних, який допоможе нейронній мережі навчитися розпізнавати стегано та звичайні сторінки. Формально цей процес поділяється на етапи:

Отримання знань про предмет дослідження - для правильного використання результатів інтелектуального аналізу даних потребується корректне розуміння проблемної області.

Вибірка даних. На цьому кроці визначається характер даних, які будуть використані. Наприклад, для стеганоаналізу цікавими є web-сторінки, які містять приховане повідомлення.

Попередня обробка даних – потребує певних дій для ідентифікації даних із попереднього етапу, а саме: очищення даних, відкидання та перетворення даних. Етап попередньої обробки даних може також включати інтеграцію даних і вибір функцій. Інтеграція даних полягає в тому об'єднанні даних, які були обрані із різних ресурсів (приклад – різні web-сторінки). Вибір функцій – визначити підмножину функцій, які підходять для отримання хорошого результату інтелектуального аналізу даних. Вибір ознак надає ряд важливих переваг, таких як:

- вибір найбільш відповідних функцій дає можливість створення бажаної моделі класифікації.
- покращення точності класифікаційної моделі.

Використання отриманих знань. Це заключний етап, у якому використовується отриманні дані. Приклад – аналіз стеганосторінки та отримання доказів використання HTML стеганографії.

2.6.2 Класифікатори

Для використання класифікатора у стеганоаналізі необхідно визначити, як саме необхідно ідентифікувати належність об'єкта до певного набору категорій (класів). Якщо використовується лише два класи - це задача двійкової класифікації. Завдання класифікації можна сформулювати у вигляді кортежу (X, Y), де X - набір атрибутів, а Y - набір класів або цільових атрибутів. Хоча набір

атрибутів може бути дискретним чи безперервним, класи повинні бути дискретними.

Класифікація має безліч застосувань, наприклад: класифікація результатів аналізу крові на наявність вірусу як позитивних так і негативних. Для класифікації спочатку необхідно побудувати класифікатор, який працює на основі попередньо позначеного навчального набору. Оцінка ефективності класифікатора проводиться із використанням маркованого тестового набору. Навчальний і тестовий набір обираються різними для більшої точності.

Існує багато алгоритмів, які можна використовувати для побудови потрібного класифікатора, проте для стеганоаналізу використана нейронна мережа.

Для отримання оцінки результату бінарного класифікатора існує ряд показників, які повинні враховуватися. Більшість із них отримані з використанням матриці помилок. Матриця помилок має форму таблиці, у якій порівнюються значення класів, передбачених класифікатором та значення справжніх спостережуваних даних. За допомогою використання лічильників True Positive (TP), True Negative (TN), False Positive (FP) і False Negative (FN) можна визначити різні виміри продуктивності класифікатора. Найбільш часто використовуваними лічильниками є Accuracy, Sensitivity, Specificity [25].

- Точність(Accuracy): Відношення кількості правильних передбачень до загального числа передбачень. Точність є загальною мірою продуктивності класифікаторів. Обчислення точності за матрицею помилок у формулі на рисунку 2.5:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Рисунок 2.5 – Формула обчислення точності

- Чутливість (Sensitivity**)** або справжня позитивна швидкість true positive rate (TPR), обчислюється за формулою на рисунку 2.6:

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

Рисунок 2.6 – Формула обчислення чутливості

- Специфічність(Specificity**)** - коефіцієнт помилкових позитивних міток негативного класу, розраховується за формулою на рисунку 2.7:

$$\text{Specificity} = \frac{FP}{FP + TN}$$

Рисунок 2.7 – Формула обчислення специфічності

Площа під кривою (AUC): AUC - це комбінація чутливості та специфічності в одній метриці. Різні пороги ({0.0, 0.01, ...}) використовуються для розрахунку і побудови чутливості і специфічності на графіку. Спеціфічність знаходиться по осі X, чутливість - по осі Y. Отримана крива відома як крива ROC. AUC - це площа під кривою ROC. Класифікатор зі значенням AUC 1,0 буде вважатися ідеальним класифікатором.

2.6.3 Стандартне відхилення порядку атрибутів

При використанні HPAT у сторінку-контейнер вносяться зміни. Ці зміни можуть бути непомітні візуально, проте відображатися статистично. Якщо слідкувати за зміною положення атрибутів у тегах і порівнювати із деяким стандартим значенням, стає можливим відрізнити звичайну та стеганосторінку. Припустимо, що положення атрибутів тегу дуже сильно відрізняється від звичайного, якщо сторінка не несе прихованого повідомлення. Стеганоаналітик може порівнювати фактичне відхилення із деяким пороговим значенням, що

полегшить задачу класифікації. Проте такий метод має декілька суттєвих проблем:

• Розраховане стандартне відхилення може відрізнятися для різних web-сторінок, які не є стеганоконтейнерами. Звідси можна зробити висновок, що встановлення деякого порогу стандартного відхилення для багатьох сторінок буде неефективним.

• У випадку оновлення сторінки (сайти новин, блоги), значення стандартного відхилення в один момент часу не буде відповідати значенню в інший.

Приклад: таблиця 2.1 з використанням тегів alt та src. У ній наведені позиції атрибутів до і після додавання прихованого повідомлення англійською мовою. Для вбудування був використаний алгоритм Хуанг-Зонг. З таблиці можна бачити, що номер позиції атрибута змінюється в результаті приховування повідомлення. Атрибут alt з'являється десять разів в положенні “0” до вбудування повідомлення. У випадку src, перед вбудуванням атрибут з'являється 11 разів. Як і з alt, положення атрибутів src змінилося після приховування повідомлення.

Таблиця 2.1 – Кількість змін позицій атрибутів при Хуанг-Зонг

Атрибут	Позиція атрибута до вбудування повідомленн	Позиція атрибута після вбудування повідомленн
alt	0000000000	0001010111
src	33333332001	32322312103

Згідно до таблиці 2.1, дисперсія позицій атрибутів збільшуватиметься при спробі вбудування повідомлення у сторінку.

У роботі використовувалася функція VattPositions() для обрахунку дисперсії позиції атрибутів a_i, що зображена на рисунку 2.8:

$$VattPositions(a_i) = \frac{\sum_{k=1}^{m} (p_k - av)^2}{m}$$

$$av = \frac{\sum_{k=1}^{m} p_k}{m}$$

Рисунок 2.8 – Функція VattPositions()

Також була використана функція обрахування VS - сумарної дисперсії позицій атрибутів для всіх тегів. Загальна дисперсія дорівнює сумі їх дисперсій, де n – загальна кількість всіх атрибутів тегів у S, формула на рисунку 2.9:

$$VS = \sum_{i=1}^{n} VattPositions(a_i)$$

Рисунок 2.9 – Формула VS

Іншими словами додавання повідомлення призведе до росту дисперсії позицій атрибутів, що в свою чергу вплине на значення стандартного відхилення. Такі зміни можуть бути успішно використані як індикатор присутності застосування алгоритмів HPAT. В статистиці стандартне відхилення є мірою, яка використовується для кількісної оцінки дисперсії набору значень даних щодо їх середнього значення. Низьке значення стандартного відхилення вказує, що розподіл значень даних близький до середнього. В свою чергу високе значення стандартного відхилення вказує, що розподіл значень даних набагато більш поширений.

Для будь-якого набору даних S стандартне відхилення можна обчислити за допомогою формули на рисунку 2.10:

$$St.D = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}}$$

Рисунок 2.10 – Формула обчислення стандартного відхилення

Де: x_i - значення даних у множині S, n - число значень даних у S, μ - середнє значення даних у наборі S.

2.7 Стеганоаналіз зміни регістру атрибутів

Підхід до виявлення HCCL ґрунтується на порушенні гладкості тегів при зміні регістру. При використанні стеганографії HCCL літери тегів будуть мати вигляд комбінації нижнього та верхнього регістрів. У роботі Huang-Sun [22] використовується концепція зміщення тегів. Зміщення тегів виступає абсолютною величиною, яка підсумовує відстані між двома суміжними літерами в тезі. Вона обчислюється з використанням функції на рисунку 2.11:

$$F(T(x_1, x_2, ..., x_n)) = \sum_{i=1}^{n-1} |A(x_{i+1}) - A(x_i)|,$$

Рисунок 2.11 – Функція обчислення зміщення тегів

Де T (x_1, x_2, ...x_n) - HTML тег. $A(x_i) \in (c^j \cup C^j)$ так, що c^j = {*коди нижнього регістру ASCII*} та C^j = {*коди верхнього регістру ASCII*}. Використання стеганографії зміни регістру призведе до збільшення показника відстані тегів, що є підставою для підозри і стеганоаналізу сторінки.

2.8 Стеганоаналіз ICS

Для виявлення стеганографії використовується ідея ймовірнісної моделі виявлення, задачою якої є відповідь на питання: були вбудовані невидимі символи в HTML файл чи ні. Для цієї мети в роботі Луо- Суі[23] були використані такі значення ймовірності виникнення: ймовірність виникнення символу пробілу та ймовірність виникнення послідовності символів пробілу чи табуляції. Перша ймовірність обчислюється з використанням рівняння, зображених на рисунках 2.12 та 2.13.

$$p_{tsco}(W) = \frac{N_{ws}}{N_{allchar}}$$

Рисунок 2.12 – Формула обчислення ймовірності виникнення символу пробілу

$$p_{scso}(W) = \frac{N_{wss}}{Nws}$$

Рисунок 2.13 – Формула обчислення ймовірності виникнення послідовності символів пробілу

Де:

- W – web-сторінка.
- $N_{allchar}$ – Кількість усіх символів у web-сторінці.
- N_{ws}- Кількість символів пробілів у web-сторінці.
- N_{wss}- Кількість послідовностей символів пробілу у web-сторінці.

Отримані ймовірності порівнювалися з попередньо визначеними порогами. Результатом цього порівняння є висновок, була проаналізована web-сторінка стеганоконтейнером чи ні. Порогові значення в цьому випадку були визначені за допомогою роботи [24], у якій обчислена середня імовірність появи символу пробілу.

Також використовувався підхід темпу вбудування - (e_{rate}). Темп вбудування представлений у вигляді співвідношенням між довжиною прихованого повідомлення M і розміром web-сторінки ($W_{allchar}$). Темп вбудування для web-сторінки e_{rate} (W) обчислюється за рівнянням на рисунку 2.14:

$$e_{rate}(W) = \frac{M}{W_{allchar}}$$

Рисунок 2.14 – Формула обчислення темпу вбудування для web-сторінки

Авторами було визначено нормальний розподіл темпу вбудування, використовуючи середнє та стандартне відхилення розподілу e_{rate} (μ і δ відповідно) для визначення порогу присутності стеганографії у сторінці.

Висновки до розділу 2

У цьому розділі було проведено детальний огляд методів стеганографії у web-сторінках, а також розкрито основні цілі та засоби цифрової стеганографії.

Огляд методів стеганоаналізу показав, що існує можливість покращення вже існуючих методів для більш ефективної протидії стеганографії. Найскладнішим для виявлення є метод HPAT, який використовує маніпуляції з позиціями атрибутів тегів для приховання повідомлення. Проблемами для виявлення HPAT є його неочевидність, оскільки застосування алгоритму до сторінки не призводить до збільшення розміру файлу чи помітних візуальних змін.

Для стеганоаналізу HPAT було обрано використання класифікатора для розподілу сторінок на порожні та заповнені стеганоконтейнери.

3 СТЕГАНОАНАЛІЗ HPAT

У попередньому розділі згадувалася ідея використання моделі класифікації для розрізнення стегано та звичайних сторінок, а також використання стандартного відхилення для ідентифікації HPAT. Проте в обох випадках не враховувалися деякі важливі показники, що могло призвести до погіршення точності виявлення стеганографічних методів.

3.1 Використання лічильника зміни позицій

Підхід використання класифікатора для стеганоаналізу HPAT був використаний у [21]. У роботі фігурувало відношення значення відстані між середніми позиціями атрибутів на web-сторінці до значення відстані між середніми позиціями атрибутів на стеганосторінці. Проте при такому підході потенціально можлива втрата деяких значень, що може критично вплинути на результат аналізу.

Замість середнього значення, для стеганоаналізу сторінки застосовується лічильник зміни позиції.

Лічильник зміни позицій використовується для підготовки даних, які будуть використані класифікатором. Для класифікації необхідне представлення даних у векторному вигляді, тому алгоритм знаходить всі зміни і перетворює в зручний для класифікатора формат. В свою чергу, класифікатор спроможний розрізнити сторінки стеганоконтейнери та звичайні, за умови застосування до них методу HPAT.

Запропонований алгоритм використовує концепцію зміни атрибутів позиції. Ідея полягає у тому, що використання HPAT до web-сторінки призводить до частих змін позицій атрибутів у тегах.

Для реалізації алгоритму визначається номер позиції кожного атрибуту в тезі. Перша позиція відповідає номеру 0, друга – номеру 1. Таким чином

нумерується позиція кожного атрибута в тезі. Тоді певний набір атрибутів матиме пов'язаний із ним набір позицій.

За відсутності HPAT очікується, що значення у наборі позицій будуть більш постійними, ніж у випадку застосування стеганографії до сторінки. Звідси можна зробити висновок, що підрахунок кількості змін позиції атрибутів допоможе виявити присутність методу HPAT. Підрахунок здійснюється шляхом порівняння послідовних позицій - якщо відбулася зміна положення, то кількість змін для обраного атрибута збільшується на 1.

Концепція лічильника зміни позиції продемонстрована у таблиці 3.1 на прикладі type та src.

Таблиця 3.1 – Зміна положень атрибутів до і після HPAT

Атрибут	Позиція атрибуту до HPAT	Лічильник зміни позицій до HPAT	Позиція атрибуту після HPAT	Лічильник зміни позицій після HPAT
type	1111111111	0	1101001010	7
src	000000000001111	1	101000110104242	11

У другому стовпці таблиці представлено можливі позиції атрибутів у web-сторінці при відсутності застосування HPAT. Перший атрибут зустрічається 10 разів, другий - 15 (відповідно 10 і 15 позицій). У третьому стовпці наведено список пов'язаних змін позицій. Четвертий стовпець демонструє можливі позиції атрибутів web-сторінці при застосуванні HPAT. П'ятий стовпець показує асоційоване число змін позицій. Спостерігається явна різниця показників лічильника до і після HPAT.

3.2 Алгоритм лічильника зміни позиції атрибутів

Ідея використання даного алгоритму полягає у використанні вилученої ним інформації для отримання вектору ознак. Це дозволить створити якісний навчальний набір, на основі якого класифікатор робить висновок про відсутність чи присутність застосування HPAT стеганографії. Навчальний набір складається із рівної кількості звичайних і стеганосторінок.

Для побудови класифікаційної, також відомої як моделі прогнозування, існує досить багато алгоритмів (також відомих як моделі прогнозування). Загальними прикладами є такі алгоритми: нейронні мережі, машини підтримки векторних зображень та NB. Спільним для них є представлення вхідних даних - в якості вхідного набору ознак обираються вектори довжини n, породжені n вимірним простором ознак. У випадку стеганоаналізу HPAT, простір ознак представлений у вигляді набору атрибутів. Кожен вимір представляє атрибут, діапазон значення якого знаходиться між 0 та невизначеним значенням кількості змін позицій.

Навчальні дані містять матрицю n × m, у якій n - кількість web-сторінок у навчальному наборі, m - кількість врахованих атрибутів. Як і під час описання попередніх алгоритмів, слід зазначити наступні визначення: H - web-сторінка, S - набір тегів T в H, які мають два чи більше атрибутів, S = {T1, T2,. . . }, FV - генерований вектор характеристик всіх позицій атрибутів тегів в S. Positions(*a*) є масивом позицій у S для атрибута ai, Positions(*a*) = [p1, p2,. . . *p*m]. *apcc*(*a*) позначено кількість змін позиції для атрибута (*a*).

Алгоритм для генерування вектора властивостей представлено на рисунці 3.1. У якості вхідних даних для роботи алгоритму виступає сторінка H. В результаті дії алгоритму отримано вектор ознак FV, у якому записана кількість змін позицій атрибутів по відношенню до всіх тегів в межах S. Робота алгоритму починається із пошуку придатних до зміни тегів S. Після цього він циклічно проходить через кожен тег T в S, перебираючи атрибути (a) в кожному тезі.

Позиція атрибуту (a) записується у масив Positions за допомогою функції collectPositions(). Потім алгоритм проходить через кожен масив Positions () для обчислення зміни позиції атрибутів. Кожне обчислене значення зберігається у FV, щоб потім сгенерувати необхідний для стеганоаналізу вектор ознак. Для навчальних даних кожен вектор ознак має асоційовану з ним мітку класу: клас 1 представляє web-сторінку стеганоконтейнер , клас 2 - звичайну web-сторінку.

```
for i in range(0,len(computer_raw_data['urls'])):
    url = computer_raw_data['urls'][i]
    opener = urllib.request.urlopen(url).read()
    soup = BeautifulSoup(opener, 'lxml')
    tags = [tag for tag in soup.find_all()]
    S=[]
    FV =[]
    for tag in tags:
        if len(tag.attrs) > 1:
            S.append(tag)
    for tag in S:
        attrPositions =[]
        for attr in tag.attrs:
            apcc=0
            attrPositions.append(attr.index)
            for p in attrPositions:
                if p == next(p):
                    apcc = apcc+1
            FV.append(apcc)
```

Рисунок 3.1 –Алгоритм HPAT

3.3 Оцінка роботи алгоритму

Для оцінки алгоритму був використаний навчальний набір даних. Перевага підходу із використанням лічильника полягає у однаковій ефективності виявлення HPAT незалежно від стеганографічного алгоритму. Однією із задач була перевірка впливу довжини вбудованого повідомлення на роботу

класифікатору, а також порівняння єфективності із існуючими алгоритмами. Для досягнення цієї цілі використані алгоритми Хуанг-Зонг та Шен, за допомогою яких у випадкову сторінку із набору вноситься стеганографічне повідомлення довжиною у 30 символів.

Для застосування HPAT обрано 6 атрибутів тегів. За таких умов вектор ознак має форму 6 динамічних слотів. У кожному із випадків матриця даних визначалася як 200 × 6, де 200 – кількість web-сторінок включених до набору. Для оцінки використовувалася десятикратна перехресна перевірка (TCV), в ході якої вхідні дані були розділені на 10 наборів. Процес тестування класифікатора відбувся 10 разів, кожен раз із використанням наступного тестувального набору. У якості метрик були обрані параметри, описані у розділі 2: відсоток вдало класифікованих екземплярів та площа під кривою.

Розглянуто дві моделі генерації класифікаторів: нейронна мережа (NN) та метод опорних векторів (SVM). Для побудови класифікатора на базі SVM використані стандартні параметри. Результати з використанням кожної метрики знаходяться у таблиці 3.2. Перший стовпчик – назва алгоритму, у наступних двох знаходяться показники, отримані в умовах метрики Accuracy. Виділені жирним стовпчиким містять показники, отримані в межах метрики AUC. З таблиці можна бачити, що запропоноване векторне представлення алгоритму лічильника перестановок атрибутів може бути успішно використано для навчання класифікаторів з метою успішного розрізнення звичайних сторінок від стеганосторінок. Найкращі показники були отримані для алгоритму HPAT Хуанг-Зонг. Дані з таблиці вказують на успішність алгоритму виявлення, незалежно від застосованого стеганографічного підходу. Також відсутня значна різниця між обраними генераторами класифікаторів.

Таблиця 3.2 - Результати для середньої точності та середнього AUC за використання лічильника перестановки атрибутів.

Алгоритми HPAT	NN	SVM	NN	SVM
Шен	92.34%	92.22%	0.96	0.95
Хуанг-Зонг	92.51%	94.46%	0.99	0.95

Окремо розглядається вплив довжини та складу повідомлення на процес його виявлення. Було обрано два тестових набори – у web-сторінки першого приховувалося звичайне повідомлення. Для другого набору повідомлення складалося із випадкового тексту обмеженої довжини, причому для формування тексту використовувалися літери верхнього та нижнього регістрів, спецсимволи і цифри.

Для кожного набору довжина повідомлення знаходилась у межах від 10% до 80% максимальної ємності сторінки контейнера. Для вбудування повідомлень використовувались алгоритми Шен та Хуанг-Зонг. Із генераторів класифікаторів були знову використані NN та SVM.

Також було виконане порівняння запропонованого підходу та підходів Полака-Котульського [21] і Хуанг-Зонга[22]. Для корректного порівняння , представлений у [21] алгоритм був адаптований через представлення значення W у вигляді вектору ознак. При такому підході кожна web-сторінка була представлена статистичною ознакою W. У випадку алгоритму 22 вектор ознак складається із запропонованих у роботі Полака-Котульського статистичних ознак - стандартних значень відхилення. Це дало змогу представити сторінку однією особливістю, тобто стандартним відхиленням позицій атрибутів.

Для оцінки знову використовувалися попередні алгоритми HPAT та генератори класифікаторів NN і SVM – 4 комбінації. Результати представлені в таблиці 3.4:

Таблиця 3.4 – Порівняння результату дії ЛЗП з іншими техніками виявлення

	Підхід до виявлення HPAT	Accuracy	AUC	Accuracy	AUC
NN	ЛЗП	91.06%	0.94	92.43%	0.97
	Полака-Котульського	70.12%	0.87	61.27%	0.55
	Луі-Шенг	78.93%	0.95	73.32%	0.80
SVM	ЛЗП	91.21%	0.9	95.27%	0.94
	Полака-Котульського	73.70%	0.73	60.10%	0.59
	Луі-шенг	79.43%	0.81	74.84%	0.75

З таблиці 3.4 можна побачити, що підхід ЛЗП здатен ефективно виявляти приховане повідомлення не залежно від використаного алгоритму стеганографії (у випадку застосування підходу HPAT). Також запропонований метод показав найкращу точність (Accuracy) у всіх випадках. А для метрики AUC - найкращий результат у 3 із 4 випадків.

3.4 Статистичне виявлення HPAT

Як було зазначено у роботі Полака-Котульського, стандартне відхилення позиції атрибутів може використовуватися як індикатор застосування HPAT у web-сторінці. На основі цього значення можливо створити деякий поріг, згідно до якого відбувається ідентифікація стеганосторінок. Описаний у 23 поріг W був використовувався для опису співвідношення пар атрибутів HTML із різним упорядкуванням тегів до загальної кількості пар HTML атрибутів. У цьому підході значення W може виступати у ролі константи або змінної, проте не були запропоновані конкретні порогові значення для W.

У представленому алгоритмі для визначення стандартного відхилення враховується формат вбудованого повідомлення. У даному алгоритмі – H позначається вхідна web-сторінка, яка є можливим стеганоконтейнером. S позначено набір тегів T в H, які мають два або більше атрибутів, S = {T1, T2, . }. Tj = {a1, a2, . . . , an} позначає j-й тег у S разом з його атрибутами. attPositions(a_i) - масив позицій у S для атрибута (a_i), a_i, $attPositions(a_i)$ = [$p1, p2, . . . pm$]. Кожен атрибут має m входжень,звідси довжина (attPositions(ai))=m.

VattPositions(a_i) - дисперсія позицій атрибута ai в масиві VattPositions(), формула для обрахунку якої описана у роботі , як і VS - сумарна дисперсія позицій атрибутів для всіх тегів у S.

SD - стандартне відхилення VS, яке використовується для визначення кількості змін позиції атрибутів web-сторінки. Обчислюється за допомогою вилучення квадратного кореню із VS.

Процес обчислення стандартного відхилення розміщення атрибутів продемонстровано на рисунку 3.2:

```
for i in range(0,len(computer_raw_data['urls'])):
    url = computer_raw_data['urls'][i]
    opener = urllib.request.urlopen(url).read()
    soup = BeautifulSoup(opener, 'lxml')
    tags = [tag for tag in soup.find_all()]
    S = []
    FV = []
    for tag in tags:
        if len(tag.attrs) > 1:
            S.append(tag)
    VS = 0
    for tag in S:
        attrPositions =[]
        for attr in tag.attrs:
            attrPositions.append(attr.index)
            m = len(attrPositions)
            VattrPos = 0
            for p in attrPositions:
                av = sum(p)/m
                VattrPos = sum((p-av)^2)/m
            VS = VS+VattrPos
    St_D=math.sqrt(VS)
```

Рисунок 3.2 – Алгоритм визначення стандартного відхилення

Алгоритм аналізує web-сторінку H, в результаті чого визначається значення стандартного відхилення позицій атрибутів у наборі тегів S. Алгоритм проходить сторінку і знаходить S перший придатний набір тегів з двома або більше атрибутами. Функції collectPositions(a_i) збирає в масив VattPositions(a_i) позиції атрибутів. Обчислення загальної дисперсії позицій атрибутів VS обчислюється із використанням послідовності позицій атрибутів у VattPositions(a_i) для кожного атрибуту. В результаті отримано відповідне значення стандартного відхилення.

У таблиці 3.5 наведено значення стандартного відхилення, які були обраховані описаним алгоритмом.

Таблиця 3.5 – Стандартне відхилення сторінок

Web-сторінка	Стандартне відхилення до вбудування повідомлення
joyreactor.cc	4.07
www.microsoft.com/ua	3.87
www.ebay.com	6.91
stackoverflow.com	3.20
www.amazon.com	6.22

Було обрано 5 сторінок, до яких було використано алгоритм знаходження стандартного відхилення. Сторінки не мали в собі прихованого повідомлення, проте за наведеними у таблиці результатами можно побачити різницю між значеннями стандартного відхилення. З таблиці можна бачити, що обчислені між вибраними web-сторінками значення SD істотно відрізняються. Це демонструє проблему визначення відповідного порогу для виявлення APS.

3.5 Стеганоаналіз із використанням стандартного відхилення

Якщо протягом певного часу підраховувати стандартне відхилення певної сторінки, то стає можливим виявлення факту використання HPAT. У вигляді прикладу можна розглянути ситуацію, за якої на певний оновлювальний ресурс в певний період часу може бути вбудовано приховане повідомлення. В такому

разі значна зміна стандартного відхилення може свідчити про використання стеганографії. Проте для цього необхідно встановити певний порог σ, опираючись на значення якого можна зробити висновок про природу web-сторінки.

Одним із варіантів отримання порга σ є його експерементальне встановленя за допомогою тестового набору сторінок із різними стеганографічними повідомленнями. Тобто до початку спостереження за ресурсом доцільно провести тренувальний процес. Обчислення σ має вигляд формули на рисунку 3.3:

$$\sigma = (\alpha \times SD_{до}) + ((1 - \alpha) \times SD_{після})$$

Рисунок 3.3 – Формула обчислення σ

α є коефіцієнтом чутливості який приймає значення від 0 до 1, $SD_{до}$ є середнім значенням стандартного відхилення до додавання повідомлення, $SD_{після}$ є середнім значенням стандартного відхилення після додавання повідомлення.

Для моніторінгу стандартного відхилення необхідні:

• H- web-сторінка з оновлюємим змістом.

• α- коефіцієнт чутливості, 0 <α <1.

• T- загальний час навчання.

• τ - Інтервал між знімками сторінки.

• L- довжина вбудованого повідомлення.

Моніторинг динамічної сторінки для зручності можна розбити на кроки:

1.Збір послідовності тренувань. Накопичується серія відбитків сторінки H з інтервалом τ протягом періоду часу T. Утворена множина $C = \{s1, s2, \ldots, sn\}$.

2. Обчислення порогу σ з використанням значення стандартного відхилення.

- Для кожного відбитку s ∈ C обчислюється стандартне значення відбитку s. Після цього у сторінку вбудовується повідомлення.
- Також до вбудування рахується середнє значення *SD,* яке також також рахується і після накладання. Середне значення в обох випадках визначається як відношення суми SD до/після до n.

n позначається кількість знімків.

3. Виявлення. Обчислюється значення SD кожної наступної копії H і порівнюється із σ сторінки. Якщо SD ≥ σ, то сторінка є стеганоконтейнером із вбудованим повідомленням. В іншому випадку web-сторінка є порожнім контейнером.

3.6 Оцінка методу стандартних відхилень

Для підтвердження ефективності виявлення HPAT методом стандартних відхилень необхідно:

- Підтвердити твердження про те, що збільшення значення SD для web-сторінки дійсно можна вважати індикатором HPAT незалежно від алгоритму вбудування.
- Порівняти значення природніх змін показника SD зі значенням SD після вбудовувавання повідомлень.
- Перевірити вплив вбудування різних форм повідомлень на запропонований механізм виявлення HPAT.
- Підтвердити ефективність SD позиції атрибута для використання у процесах стеганоаналізу сторінок.

Для підтвердження ефективності методу SD у виявленні HPAT було обрано набір із відбитків трьох сторінок: Github, Habr, Вікіпедія. У кожному випадку для приховування повідомлення було використано стеганографічний метод перестановки атрибутів – Хуанг-Зонг і Шен.

Для вбудування були обрані повідомлення різної довжини L, які використовували від 20% до 100% максимальної ємності сторінки. При кожному кроці довжина повідомлення збільшувалась на 10%. Максимальна ємність для сторінок Github, Habr та Вікіпедія становила 70, 74 і 117 байт відповідно.

Результати зображені на Рисунку 3.4:

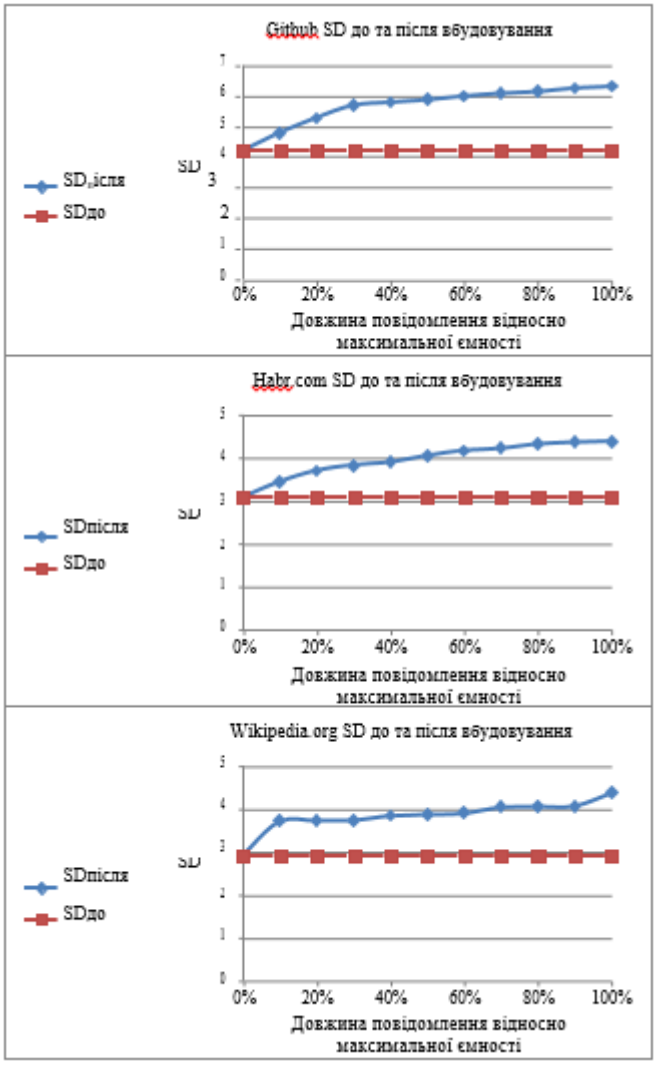

Рисунок 3.4 – Відношення стандартного відхилення до довжини повідомлення

Віссю *x* представлена довжина повідомлення відносно максимальної ємності сторінки. Вісь *y* – значення SD до і після додавання повідомлення. З рисунку можна помітити, що SD збільшується по відношенню до SD для обраної

сторінки без будь-якого прихованого повідомлення через збільшення розміру вбудованого повідомлення. Можна зробити висновок, що використання SD як індикатора стеганографії є ефективною методикою.

На рисунку 3.5 показано результати із урахуванням довжини повідомлення і використаного алгоритму вбудування. Графіки з лівої сторони відносяться до вбудованих за алгоритмом Хуанг-Зонг повідомлень, графік праворуч - Шен.

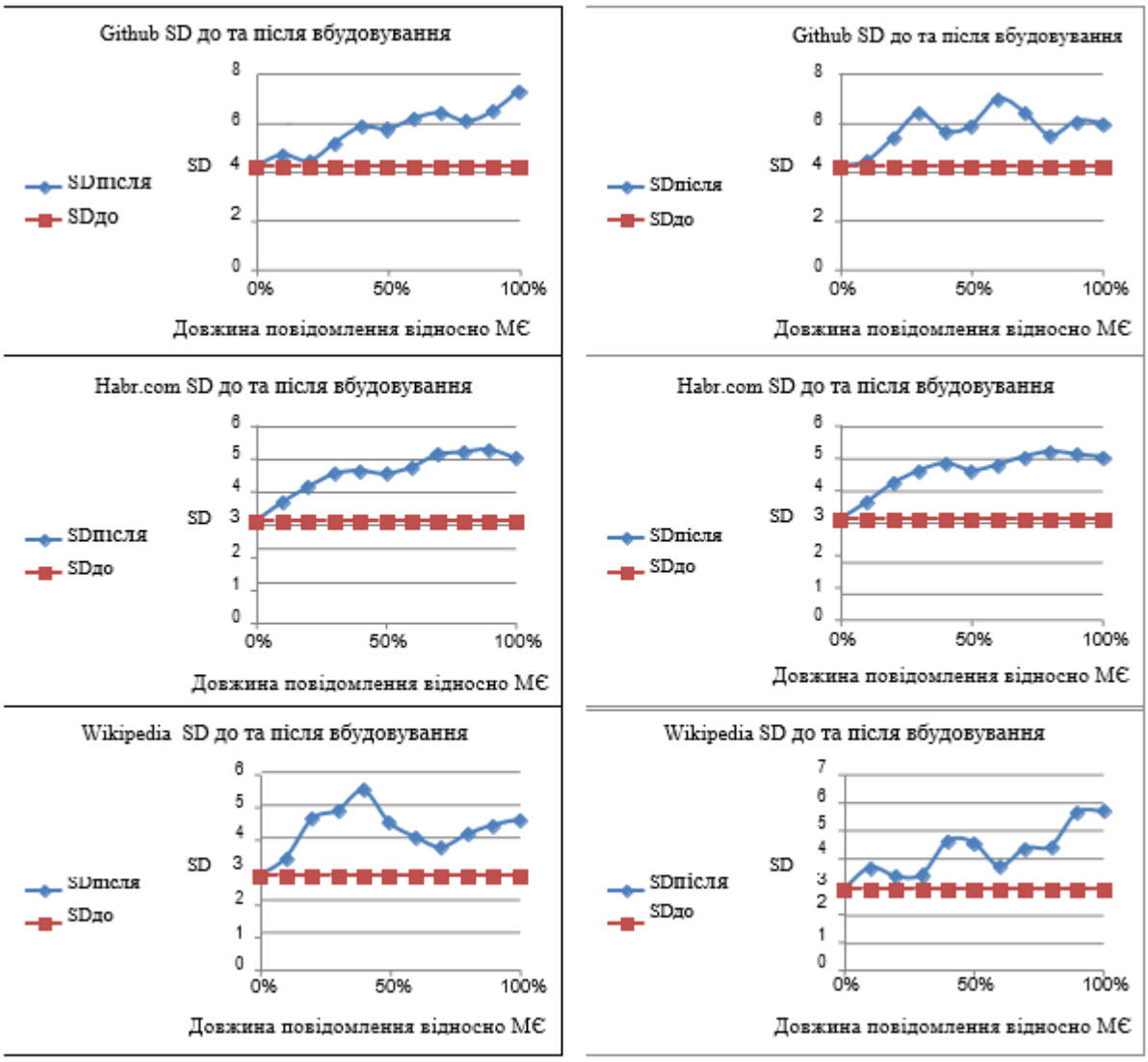

Рисунок 3.5 – Графік стандартного відхилення при застосуванні алгоритмів Хуанг-Зонг та Шен

На рисунку 3.5 підтверджено ефективність використання стандартного відхилення для виявленя стеганосторінок незалежно від застосованого до них алгоритму вбудуваня.

Попередні графіки демонструють значення SD без урахування впливу формату повідомлення на результат. Для урахування цієї особливості був використаний той самий набір сторінок, проте вбудовувалися різні формати повідомлень – випадковий текст, комбінації літер нижнього і верхнього регістрів, цифри. Для цього було використано 200 різних повідомлень, серед яких було 20 повідомлень довжини від 10% до 100% ємності сторінки. Результати представлені в таблиці 3.6 У таблиці рядок «Середнє» показує середнє значення SD для сторінки після вбудування. Рядок «Відхилення» відображає значення стандартного відхилення для SD після вбудування збільшених повідомлень.

Таблиця 3.6 – Усереднені значення SD з використанням різноформатних вбудованих повідомлень

	Довжина вбудованого повідомлення									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Середнє	6.03	6.15	6.40	6.44	6.55	6.80	6.83	6.91	6.83	7.25
Відхилення	0.05	0.04	0.13	0.11	0.20	0.13	0.12	0.15	0.18	0.14

З результатів таблиці можна зробити висновок про відсутність впливу формату прихованого повідомлення на можливість виявлення.

Ще однією ціллю оцінки було порівняння природнього значення SD сторінки та значення при додаванні повідомлення. Природнє значення змінюється через часте оновлення таких сторінок як прогнози погоди чи сайти новин. Набір даних для порівняння було побудовано з використанням параметрів: T – 2 дні, τ – кожні 1.5 години. Довжина вбудованого повідомлення

становить 30% загальної ємкості сторінки. Розрахунок SD проводився до і після приховування повідомлення.

Результати представлені на рисунку 3.6:

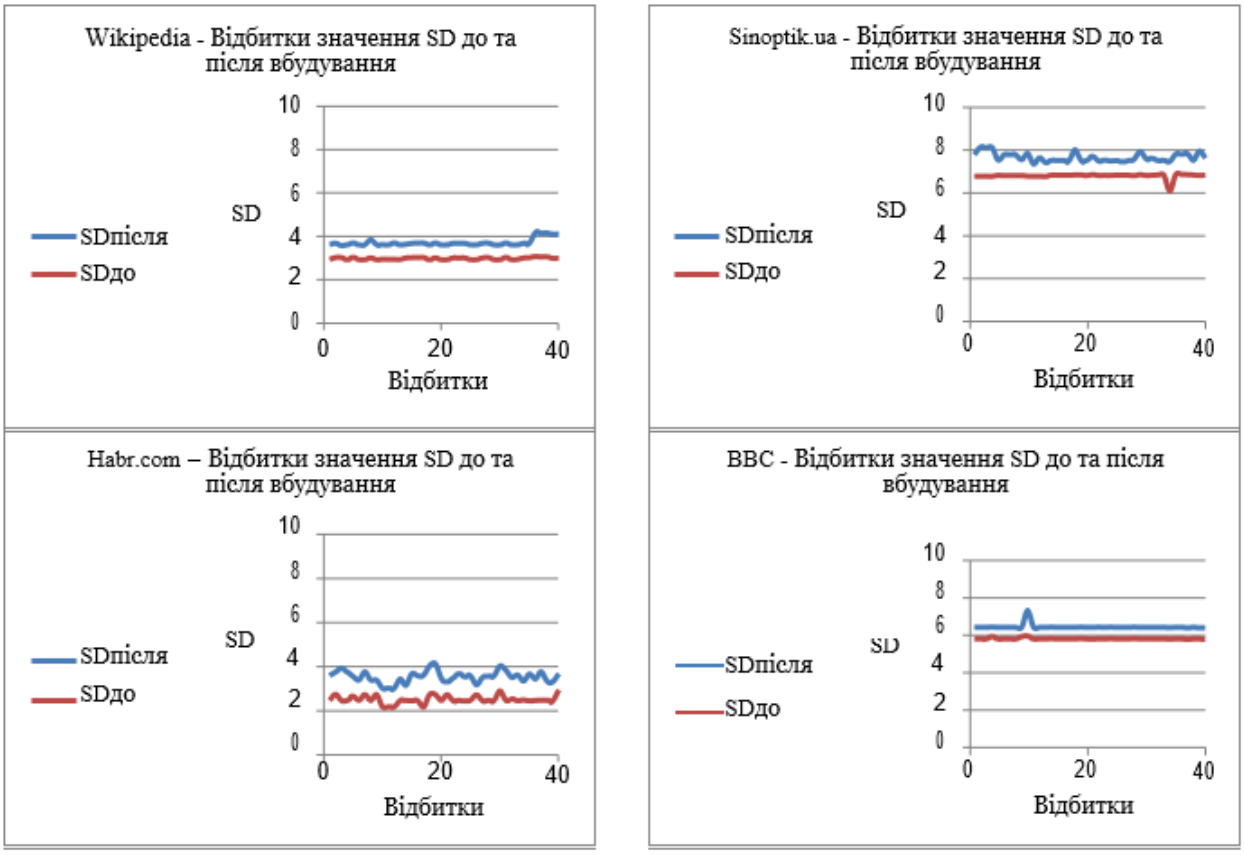

Рисунок 3.6 – Значення SD до і після вбудування повідомлення

Віссю *x* представлено номер кожного відбитку сторінки, віссю *y* - значення SD для відповідної сторінки. З рисунку можна побачити значну різницю між значеннями SD у сторінок з/без використання стеганографії. Повідомлень. Зміни що відбуваються природно, істотно відрізняються. Звідси можна зробити висновок про ефективність використання стандартного відхилення для розрізнення стеганосторінок від звичайних.

Висновки до розділу 3

Описані підходи показали свою ефективність у виявлення застосування HPAT. Класифікатор на основі алгоритму лічильника зміни кількості позийцій атрибутів у більшості випадків вдало виявляє стеганосторінки. Від представлених раніше методів даний класифікатор відрізняється більшим відсотком корректно виявлених сторінок. Запропонований алгоритм дозволяє формувати точний дата-сет для навчання нейронної мережі або SVM.

Також вдалося довести користь визначення стандартного відхилення для ідентифікаціїї HPAT.

ВИСНОВКИ

Сьогодні стеганографія розвивається швидкими темпами. Велика кількість потенційних контейнерів інформації дозволяє розробляти нові алгоритми та методики, повязані із використанням обраного формату. Проте разом із стеганографією відбувається стрімка еволюція стеганоаналізу.

Web-сторінки продемонстрували свій потенціал та надійнійсть як стеганоконтейнери. Для них було розроблено багато алгоритмів приховання повідомлень, які використовували комбінацію особливостей формату і деяких лінгвістичних методів. Найвідоміші із способів приховання повідомлень було розглянуто у цій роботі. Разом із цим розглядалися стеганоаналітичні алгоритми та підходи, які здатні ефективно виявляти застосування стеганографії.

Більш детально розглянута стеганографічна методика, яка використовує зміну позицій атрибутів для приховання повідомлення. Оскільки вона має найбільший потенціал, було запропоноваано два методи для її виявлення. Обидва методи показали свою ефективність у розрізнені звичайних та стеганосторінок. Один із методів базувався на запропоновоному алгоритмі підрахунку змін позицій атрибутів та його використанні для побудови класифікатора. Інший метод використовавав статистичні дані про зміни, що допомогло встановити певний поріг змін.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Стеганография [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://ru.bmstu.wiki/Стеганография#.D0.9A.D0.BE.D0.BD.D1.82.D0.B5.D0.B9.D0.BD.D0.B5.D1.80.
2. Компьютерная стеганография [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/anisimovkhv/learning/kripto/lecture/tema19#p193.
3. FORENSICS AND PIRACY DETERRENCE [Електронний ресурс] – Режим доступу до ресурсу: http://digitalwatermarkingalliance.org/digital-watermarking-applications/forensics-and-piracy-deterrence/.
4. XUAN Z. STEGANOGRAPHIC FILE SYSTEM [Електронний ресурс] / ZHOU XUAN – Режим доступу до ресурсу: http://www.l3s.de/~zhou/Publication/StegFS-thesis.pdf.
5. Principles and Overview of Network Steganography [Електронний ресурс] – Режим доступу до ресурсу: https://arxiv.org/ftp/arxiv/papers/1207/1207.0917.pdf..
6. ОСНОВНЫЕ ПОЛОЖЕНИЯ СТЕГАНОГРАФИИ [Електронний ресурс] // Защита информации. Конфидент. – 2000. – Режим доступу до ресурсу: http://citforum.ck.ua/internet/securities/stegano.shtml.
7. LSB стеганография [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: https://habr.com/ru/post/112976/.
8. Стеганографический метод Куттера-Джордана-Боссена [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: https://habr.com/ru/post/115287/.
9. Implementation of LSB Steganography and its Evaluation for Various File Formats [Електронний ресурс] – Режим доступу до ресурсу: https://pdfs.semanticscholar.org/3dce/b6307cee042b687b7f377ec1d5de91ce20b0.pdf
10. Data Hiding using Graphical Code based Steganography Technique

[Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/282403473_Data_Hiding_using_Graphical_Code_based_Steganography_Technique.

11. An Overview of Steganography for the Computer Forensics Examiner [Електронний ресурс] – Режим доступу до ресурсу: https://www.garykessler.net/library/fsc_stego.html.

12. On The Limits of Steganography [Електронний ресурс] – Режим доступу до ресурсу: https://www.cl.cam.ac.uk/~rja14/Papers/jsac98-limsteg.pdf.

13. Steganography and Digital Watermarking: a global view [Електронний ресурс] – Режим доступу до ресурсу: http://lia.deis.unibo.it/Courses/RetiDiCalcolatori/Progetti00/fortini/project.pdf.

14. CURRENT TRENDS IN STEGANALYSIS: A CRITICAL SURVEY [Електронний ресурс] – Режим доступу до ресурсу: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.331.3139&rep=rep1&type=pdf.

15. Стеганография и пробел нулевой длины [Електронний ресурс] – Режим доступу до ресурсу: https://bolknote.ru/all/3556/.

16. Unicode Steganography with Zero-Width Characters [Електронний ресурс] – Режим доступу до ресурсу: https://330k.github.io/misc_tools/unicode_steganography.html.

17. Data Hiding Techniques in Windows OS H. Rami..

18. Sui X. A new steganography method based on hypertext / X. Sui, L. Hui. // 2004. – С. 181–184.

19. Zhao H. A novel scheme of webpage information hiding based on attributes [Електронний ресурс] / Hong Zhao – Режим доступу до ресурсу: https://www.researchgate.net/publication/224212217_A_novel_scheme_of_webpage_information_hiding_based_on_attributes.

20. Detection of HTML Steganography Based on Statistics and SVM Classification [Електронний ресурс] – Режим доступу до ресурсу: http://xwxt.sict.ac.cn/EN/abstract/abstract2372.shtml

21. Sending hidden data through www pages: detection and prevention [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/268076703_Sending_hidden_data_through_www_pages_detection_and_prevention.

22. Steganalysis of Information Hidden in Webpage Based on Higher-order Statistics [Електронний ресурс] – Режим доступу до ресурсу: https://ieeexplore.ieee.org/document/4606211.

23. Hui L. A steganalysis method based on the distribution of space characters / L. Hui, X. Sui. – 2014.

24. Modeling and Managing Content Changes in Text Databases [Електронний ресурс]. – 2005. – Режим доступу до ресурсу: http://www.cs.columbia.edu/~gravano/Papers/2005/icde2005.pdf.

ДОДАТКИ

ДОДАТОК А

classificator.py

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pylab import rc, plot
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import precision_recall_curve, classification_report
from sklearn.model_selection import train_test_split

df = pd.read_csv('../../data/telecom_churn.csv')


d = {'Yes' : 1, 'No' : 0}

df['International plan'] = df['International plan'].map(d)
df['Voice mail plan'] = df['Voice mail plan'].map(d)
df['Churn'] = df['Churn'].astype('int64')

le = LabelEncoder()
df['State'] = le.fit_transform(df['State'])

ohe = OneHotEncoder(sparse=False)

encoded_state = ohe.fit_transform(df['State'].values.reshape(-1, 1))
tmp = pd.DataFrame(encoded_state,
                   columns=['state ' + str(i) for i in range(encoded_state.shape[1])])
```

```
d = {'Yes' : 1, 'No' : 0}
X = df.drop('Churn', axis=1)
y = df['Churn']

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y,  test_size=0.33, random_state=42)

lr = LogisticRegression(random_state=42)
lr.fit(X_train, y_train)


def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

font = {'size' : 15}

plt.rc('font', **font)

cnf_matrix = confusion_matrix(y_test, lr.predict(X_test))
plt.figure(figsize=(10, 8))
plot_confusion_matrix(cnf_matrix, classes=['Non-churned', 'Churned'],
                      title='Confusion matrix')
plt.savefig("conf_matrix.png")
plt.show()
```